ACCEDER A UNE BASE DE DONNEES ACCESS AVEC DAO

NIVEAU : PREMIERE RENCONTRE AVEC VB INITIES/EXPERIMENTES

Pré requis pour comprendre ce tutorial :

- Connaître les principales commandes de VB
- Connaître la structure générale d'une base de données Access

Auteur : Dark sidious
Date de création : 14/06/2004
Version : 1.0

SOMMAIRE

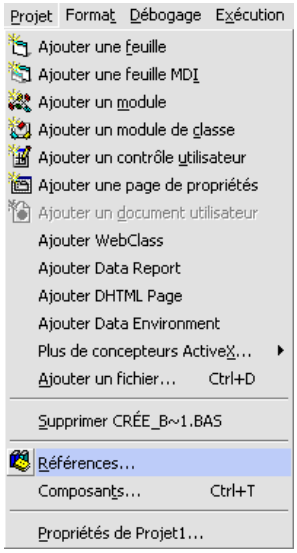
INTRODUCTION

Ce tutorial est destiné à vous apprendre à accèder à une base de données Access avec du code uniquement en utilisant la technologie DAO de Microsoft. Vous allez apprendre à récupérer des valeurs dans une base de données, les modifier, vous déplacer dans les tables et les champs de la base.
Pour avoir des informations supplémentaires sur les bases de données, si vous voyez des erreurs, si vous avez des problèmes ou que vous ne comprenez pas trop bien ce que j'essaie de vous apprendre dans ce tutorial, ou si vous avez des idées pour un nouveau tutorial, vous pouvez m'envoyer un message privé sur le site www.ProgOtoP.com à DARK SIDIOUS.

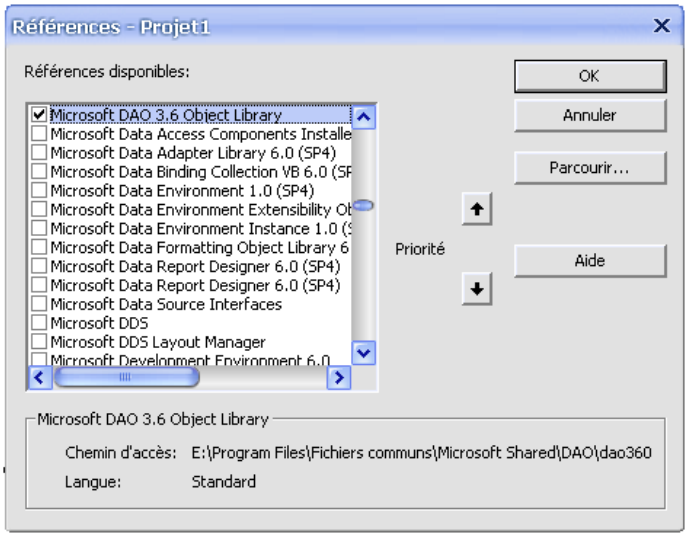
Bonne lecture, et bonne compréhension.

Les références à DAO

Avant de se lancer dans l'accès à la base de données, il vous faut définir quelques références à DAO (DataBase Object Library) pour que votre projet puisse accèder à la base avec DAO. Pour ce faire, créer un nouveau projet, de type EXE Standard par exemple. Ensuite, cliquez sur le menu Projet, puis sur Références :



Ensuite, une boite de dialogue s'ouvre vous montrant toutes les références qu'il est possible de faire appel depuis VB, rechercher la référence « Microsoft DAO 3.6 Object Library » et cochez-là :



Remarque : il est possible que vous ne possèdiez pas la même version de DAO, mais ce tutorial devrait être compatible pour toutes les versions de DAO. Si la version 3.6 n'est pas référencée par VB, cochez une autre version de DAO.

Validez alors la nouvelle référence de votre projet en cliquant sur le bouton « Ok » de la boite de dialogue.

Cette étape permet donc à VB de reconnaître la librairie ADO et vous pourrez alors l'utiliser pour accéder à la base de données, déclarer des Database et des Recordset dans votre code, etc.

Ouvrir la base de données

Pour accéder à une base de données, il faut au préalable l'ouvrir. Pour ce faire, il faut déclarer une variable qui stockera l'objet de base de données :

```
Dim DAT_Base As DataBase
```

A partir de cet objet, nous pouvons ouvrir la base de données en utilisant la méthode OpenDataBase de l'objet DBEngine de DAO. Dans cet exemple, nous utiliserons une base de données situées dans le répertoire du programme et qui s'appelle « base.mdb ». Cette base de donnée doit être protégée par un mot de passe qui est : « TestMDP » (cette base correspond à la base de données crée avec le tutorial étape par étape de création de base de données qui est téléchargeable sur www.ProgOtoP.com) :

```
Set DAT_Base = DBEngine.OpenDataBase(App.Path & "\base.mdb", False, False, "MS
    Access;PWD=TestMDP")
```

Une fois que la base de données est ouverte, vous pouvez accéder aux données qu'elle contient par l'intermédiaire d'un recordset.

Création d'un recordset

Pour pouvoir lire des données, les modifier ou se déplacer dans les tables, il faut créer un objet recordset. Un objet recordset contient les données d'une table selon une requête SQL effectuée sur la table. Il peut s'agir des données d'une table complète, uniquement certains champs de la table, ou même uniquement certaines valeurs de champs.

Pour créer un recordset, il faut tout d'abord le déclarer, puis le créer en appelant la fonction OpenRecordset de l'objet DataBase. Dans cet exemple, nous allons créer un objet recordset qui contiendra les données de toute la table noms :

```
Dim RCD_Record As Recordset

Set RCD_Record = DAT_Base.OpenRecordset(« SELECT * FROM noms »)
```

Avec un tel recordset, vous avez accès à toutes la base grâce à la requête SQL « SELECT * FROM noms ». Si à la place de l'étoile, vous marquez « prénom » par exemple, cela vous créera un recordset qui ne contiendra que les données du champ prénom de la table. Vous pouvez ainsi définir les champs dont vous voulez accéder ou non. Il est également possible d'effectuer une recherche avec une requête SQL pour ne récupérer par exemple que les prénoms qui s'appellent « Pierre » :

```
Set RCD_Record = DAT_Base.OpenRecordset(« SELECT prenom FROM noms WHERE
    prenom = 'Pierre'»)
```

Vous pouvez également definer un recordset pour récupèrer tout les champs de la table ou le prénom est "Pierre" :

```
Set RCD_Record = DAT_Base.OpenRecordset(« SELECT * FROM noms WHERE prenom
    = 'Pierre'»)
```

Enfin, il est également possible de faire de définir des recordsets qui sont le résultat d'une recherche où les valeurs ressemblent à une chaîne, par exemple :

```
Set RCD_Record = DAT_Base.OpenRecordset(« SELECT prénom FROM noms WHERE
    prenom LIKE 'P*e'»)
```

Ce recordset contiendras les valeurs du champ prénom dans lesquelles le prénom commencera par un P majuscule et se terminera par un e minuscule.

Se déplacer dans le recordset

Lorsque le recordset stocke plusieurs données d'un champ, il est courant de s'y déplacer pour récupérer ou mettre à jour toutes les valeurs du recordset. Pour cela, DAO nous propose des fonctions pour ce déplacer, pour savoir si on se trouve au début ou à la fin du recordset, le nombre d'enregistrements, etc.

Le nombre d'enregistrement est récupérable par l'intermédiaire de la propriété RecordCount du recordset. Cependant, celle-ci n'est valide que lorsque vous vous trouvez à la fin du recordset car celle-ci renvoie la position actuelle du pointeur dans la base.

Pour se déplacer dans la base, vous pouvez faire appel aux méthode MoveFirst (qui vous renvoie au début du recordset), MoveLast (qui vous renvoie à la fin du recordset), MovePrevious (qui vous renvoie une position avant la position courante), MoveNext (qui vous renvoie une position après la position courante) ou Move (qui vous permet de vous déplacer à l'endroit que vous voulez dans la base).

Vous pouvez facilement savoir si vous vous trouvez au début ou à la fin des enregistrement du recordset en récupérant la valeur des propriétés BOF (début des enregistrements) ou EOF (fin des enregistrements) : c'est propriété sont égale à True si vous vous trouvez respectivement en début ou en fin des enregistrements du recordset, sinon, elles sont égales à False.

Récupèrer/définir des valeurs de la base

Pour récupérer des valeurs dans la base de données, il faut se déplacer à l'enregistrement dont vous voulez récupérer la valeur, puis il faut utiliser l'objet Field contenu dans l'objet Recordset qui contient le champ dont vous voulez récupérer la valeur. Puis il faut utiliser la propriété Value de l'objet Field pour récupérer la valeur : par exemple, pour récupérer la valeur de l'enregistrement courant contenu dans le champs « Prénom » du recordset RCD_Record, il faut taper :

```
Dim STR_Valeur As String

STR_Valeur = RCD_Record.Field(« Prénom »).Value
```

Pour définir une valeur de la base de données, c'est un peu plus compliqué : il faut tout d'abord édité la base de données pour pouvoir modifier les valeurs en appelant la méthode Edit de l'objet Recordset, puis vous pouvez modifier la valeur en appelant la propriété Value de l'objet Field, et enfin, il faut mettre à jour le recordset en appelant la méthode Update de l'objet Field :

```
RCD_Record.Edit

STR_Valeur = RCD_Record.Field(« Prénom »).Value = STR_Valeur

RCD_Record.Update
```

Ajouter/supprimer des valeurs dans la base

Pour ajouter ou supprimer des valeurs dans la base de données, il faut tout d'abord se déplacer à l'enregistrement à partir duquel vous voulez ajouter des valeurs, ou celui dont vous voulez supprimer.

Pour ajouter une valeur, il vous suffit d'appeler la méthode AddNew de l'objet Recordset qui ajoutera alors un nouvel enregistrement au recordset à l'endroit où vous vous situez dans la base. Vous pouvez ensuite définir la valeur de ce nouvel enregistrement comme nous l'avons vu plus haut, et enfin, il faut mettre à jour le recordset en appelant la méthode Update de l'objet Recordset pour que le nouvel enregistrement soit pris en compte :

```
RCD_Record.AddNew

RCD_Record.Fields(« Prénom »).Value = « Pierre »

RCD_Record.Fields("Nom").Value = "Dupond"

RCD_Record.Update
```

Pour la suppression d'un enregistrement, c'est la méthode Remove de l'objet Recordset qu'il faut appeler à l'endroit de l'enregistrement à supprimer, puis il faut mettre à jour le recordset en appelant la méthode Update :

```
RCD_Record.Delete

RCD_Record.Update
```

Effectuer des recherches dans un recordset

Comme nous l'avons vu, il est possible de faire une sélection des enregistrements dès la création d'un recordset grâce à une requête SQL, mais il est possible (et c'est souvent le cas) que le résultat d'une requête SQL comporte plusieurs enregistrements. Il est donc possible d'effectuer une recherche sur les enregistrements contenus dans le recordset.

4 méthodes de l'objet Recordset permettent d'effectuer des recherches, il s'agit de :
- FindFirst qui cherche dans un champ la première valeur correspondant au critère en partant du début des enregistrements du recordset.
- FindLast qui cherche dans un champ la dernière valeur correspondant au critère.
- FindNext qui cherche dans un champ la première valeur correspondant au critère en partant de la position de l'enregistrement courant.
- FindPrevious qui cherche dans un champ la première valeur correspondant au critère en remontant dans les enregistrements à partir de la position de l'enregistrement courant.

Le critère est une chaîne de caractère qui définit le nom du champ avec la valeur à trouver. Si la valeur à trouver est une chaîne de caractère, il faut que la chaîne à trouver soit encadrée d'apostrophes :

```
RCD_Record.FindFirst « Prénom = 'Pierre ' »

RCD_Record.FindNext « Age = 10 »

STR_Nom = "Pierre"

RCD_Record.FindLast "Prénom = '" & STR_Nom & "'"
```

Vous avez un moyen de savoir si une recherche a aboutie ou non (si la méthode a trouvé un élément correspondant au critère) en appelant la propriété NoMatch : celle-ci vaut True si aucun enregistrement n'a été trouvé, sinon, elle vaut False :

```
RCD_Record.FindFirst « Prénom = 'Pierre ' »

If RCD_Record.NoMacth Then

MsgBox "Aucun enregistrement n'a été trouvé"

End If
```

CONCLUSION

Comme vous avez pu le voir, l'accès aux données d'une base de données se fait grâce à l'objet recordset qui stocke le contenu d'une table selon un critère de sélection qui est défini par une requête SQL. Cela simplifie grandement la programmation de n'avoir qu'à passer par un seul objet pour définir ou récupérer les valeurs d'une base, car à partir d'une seule variable, vous pouvez commander toute la table de données (si elle contient plusieurs tables, il vous faudra bien sûr plusieurs recordset).
J'espère que la lecture de ce tutorial vous a satisfaite, et que vous continuerez à programmer sous Visual Basic. Et si vous avez des commentaires à me faire, n'hésitez pas à m'envoyer vos messages sur le site www.ProgOtoP.com